



How to Configure Linux Device Mapper Multipathing (DM-MP) for Dot Hill Storage Arrays

by Steve Cellum

Revision History

Author	Date	Changes
Steve Cellum	09/26/2006	Rev 1 – Original Document
Cullen Owen	04/04/2007	Rev 2 – Added TOC. Minor updates.
Cullen Owen	04/11/2009	Rev 3 – Updates for ULP
Cullen Owen	05/20/2009	Rev 4 – Updates on Inquiry Names and recommendations

Contents

1	Scope	4
2	Quick Start.....	4
3	Introduction	5
3.1	Features of DM-MP	5
3.2	Software components of DM-MP	6
4	Theory of Operation.....	6
4.1	Terms and Concepts	7
5	Installing DM-MP	8
5.1	RedHat Linux 4.....	8
5.1.1	Multipath Setup.....	8
5.2	RedHat Linux 5.....	9
5.3	SUSE Linux 9	9
5.4	SUSE Linux 10	9
6	Configuration	9
6.1	Configuring the DotHill storage array for Multipathing	9
6.2	Configuring DM-MP with the <code>multipath.conf</code> file.....	10
6.3	Multipath.conf (Additional Detail).....	11
6.3.1	System defaults (defaults).....	11
6.3.2	Blacklisted devices (<code>devnode_blacklist</code>).....	12
6.3.3	Per-storage-array (devices)	12
6.3.4	Per-device settings (<code>multipaths</code>).....	13
6.3.5	Unified Lun (ULP) settings (devices)	14
6.4	Multipath.conf (Recommended DotHill Storage Settings)	14
6.4.1	SANnet II.....	14
6.4.2	R/Evolution 2730 / 5730.....	15
6.4.3	R/Evolution 5730 (with ULP)	15
6.4.4	R/Evolution 2330.....	16
6.4.5	R/Evolution 2330 (with ULP).....	16
6.4.6	R/Evolution 2530.....	17
7	Configuring Host Bus Adapters	17
7.1	Configuring QLogic HBA parameters	18
7.2	Configuring QLogic/Emulex HBAs for R/Evolution storage	18
8	Managing DM-MP Devices	18
9	Known Issues.....	20
10	Appendix A - <code>multipath.conf.annotated</code> file.....	22

1 Scope

Device Mapper Multipath (DM-MP) allows Linux systems running a 2.6 kernel version to route I/O over multiple paths to a storage array. DM-MP directs the I/O operations over all the usable paths as paths fail and are repaired without any interruption to the application layer.

This document explains how to configure DM-MP for Dot Hill SANnet II storage arrays and R/Evolution storage arrays with the following versions of Linux: Red Hat Enterprise Linux 4, RedHat Enterprise Linux 5, Novell SUSE 9 and Novell SUSE 10.

2 Quick Start

This section contains a concise description of the steps necessary to set up DM-MP for the SANnet II FC, SANnet II SATA or R/Evolution storage arrays. Experienced Linux systems administrator are invited to dive right in. Others can use it as an overview of the configuration process.

1. Install the DM-MP package if it is not already installed.
(See “Installing DM-MP”)
2. By default all devices are blacklisted (disqualified from control by DM-MP). Comment out or delete the following three lines that appear near the beginning of the `/etc/multipath.conf` file:

```
# devnode_blacklist {  
#     devnode "*"   
# }
```

3. Edit the devices section of the `/etc/multipath.conf` file to look like the following (two examples of DotHill storage are shown):

Example #1:

```
devices {  
    device {  
        vendor            "DotHill"  
        product           "SANnet II FC"  
        path_grouping_policy multibus  
        getuid_callout     "/sbin/scsi_id -g -u -s /block/%n"  
    }  
}
```

Example #2:

```
devices {
    device {
        vendor          "DotHill "
        product         "R/Evo 2730-2R"
        path_grouping_policy  multibus
        getuid_callout   "/sbin/scsi_id -g -u -s /block/%n"
    }
}
```

4. Preview your multipath devices by running the following command:

```
# multipath -v2 -d
```

5. Configure the system startup scripts to start the multipathd daemon at boot time:

```
# chkconfig --add multipathd
# chkconfig multipathd on
```

6. Create your devices and start DM-MP with the two commands below. (This step is only required the first time; on reboot it will be performed as a part of system startup.)

```
# multipath
# /etc/init.d/multipathd start
```

7. Your multipath virtual devices should now appear in the `/dev/mapper` directory ready for use with names of the form `mpathn`.

3 Introduction

The Device Mapper is a new infrastructure in the Linux 2.6 kernel that provides a way to create virtual block devices. These Device Mapper devices support striping, mirroring, snapshots, concatenation and multipathing. In this document we will discuss only the multipathing feature. Device Mapper block devices are also supported by the Logical Volume Manager (LVM2). DM-MP depends upon the combination of the DM-MP kernel modules and the multipath-tools user-space package.

3.1 Features of DM-MP

DM-MP provides the following features:

- **Unified View:** Provides a single view of a storage resource

- **Alternate Paths:** Redundant physical connectivity between a host system equipped with multiple HBAs and the SANnet storage array
- **Failover:** Constantly monitors each path and automatically reroutes I/O operations over a functioning alternate path upon the failure of a path
- **Failback:** Optionally moves the I/O operations back to a path that had failed but has been repaired
- **Transparency:** Implements failover or failback transparently without disrupting higher-level applications
- **Monitoring:** Notifies the user if there is a change in the path status
- **Load Balancing:** Distributes the I/O operations among all available paths to enhance performance
- **Flexible Management:** Uses a CLI with a rich choice of display options to configure and manage multipath features
- **User-Friendly Names:** Provides an option for customized names of the DM-MP devices
- **Naming Persistence:** Names of DM-MP devices persist across reboots even if there are changes in the attached storage configuration

This first release of Device Mapper is intended for use in test environments and for applications that are not mission-critical.

3.2 Software components of DM-MP

There are four software components of DM-MP:

dm-multipath kernel module - This module routes I/O among paths and path groups.

multipath configuration tool - This command configures, lists, and removes multipath devices. The command is run in `rc.sysinit` during startup and by `udev` whenever a new block device is added. It can also be used to interactively modify the attributes of multipath devices.

multipath daemon - This daemon monitors paths by checking to see if faulty paths have been repaired. When paths come back to life, the multipath daemon may also initiate changes to ensure that the optimal path group is being used.

kpartx utility - This utility creates DM-MP devices for partitions on a device. You must use this command for MS-DOS based partitions.

4 Theory of Operation

When there are multiple paths to a storage array each path appears as a separate block device. This results in multiple block devices that refer to the same storage resource or logical unit (LU). For example, if a single LU is presented through two ports in a storage array which is connected to a host with two HBAs through a single unzoned Fibre Channel switch, the host will see four

block devices: /dev/sda, /dev/sdb, /dev/sdc, and /dev/sdd. All of these devices refer to the same underlying storage resource.

DM-MP creates a single virtual block device, e.g., /dev/mapper/mpath1, that can be used by applications. DM-MP routes I/O for that virtual device through the four underlying physical devices. If one path fails, DM-MP routes uses the remaining paths to maintain connectivity between the host and the storage.

4.1 Terms and Concepts

Path – The connection from an HBA to a storage controller. Each path appears as a separate block device. A path can be in one of two states: **ready** or **faulty** depending up on whether or not it is able to handle I/O requests.

Path Group – A collection of paths. For Dot Hill storage arrays a path group can be in one of two states: **active** (the path group is currently receiving I/O requests) or **enabled** (the path group to try if the active path group has no paths in the ready state). Only one path group, the **active** path group, receives I/O at any time. Within an **active** path group, DM-MP selects the path to be used for I/O in a round-robin fashion.

Multipath Device – A multipath device is the virtual device created by DM-MP. A multipath device can be identified by either its WWID or its **alias**. A multipath device has one or more path groups.

Blacklisted Device – The user may wish to prevent some storage resources from coming under the control of DM-MP. These are called **blacklisted devices**, and they can be specified in the configuration file.

Path Priority – Each path can have a priority assigned to it by a callout program. Path priorities can be used to group paths by priority and change their relative weights for the round-robin path selector.

Path Group Priority – Each path group has a priority that is equal to the sum of the priorities of all the **ready** paths in that group. By default, DM-MP tries to ensure that the path group with the highest priority is always in the **active** state.

Failover – When I/O to a path fails, DM-MP tries to shift I/O operations to an **enabled** path group. If there are no **enabled** path groups, SCSI error return codes of 10000 will be displayed stating no device is found. I/Os may eventually timeout until a valid path is **enabled**.

Failback – At regular intervals the multipath daemon, `multipathd`, checks the priority of all path groups. If the **active** path group is not the highest priority path group, `multipathd` acts according to the failback mode. By default it immediately switches to the highest priority path group. Other options for failback are to (a) wait a user-defined length of time and then switch, or (b) do nothing and wait for manual intervention. Failback can be forced manually at any time by running the multipath command.

WWID – The WWID (World Wide Identifier) is an identifier for a multipath device that is guaranteed to be globally unique and unchanging. It is obtained from the storage array by the `getuid`

callout program. For the SANnet storage arrays it is constructed from the Logical Unit Device Identifier on the device identification page (page 0x83 of the vital product data pages).

Alias – An alias is the name that DM-MP uses in displaying information about a multipath device. By default, the **alias** is set to the WWID of the device. However, by enabling the “user_friendly_names” option in the `multipath.conf` file, the **alias** is set to a name of the form `mpathn`. While this form of the alias is guaranteed to be unique on a host, it may not be the same on all hosts using that multipath device, and it may change. The association between user-friendly names and their WWIDs is cached in the file `/var/lib/multipath/bindings`. This file should usually not need to be modified by hand. To ensure that the multipath devices have the same names on all hosts accessing them this file can be propagated to all the hosts.

The **alias** can optionally be set to any arbitrary value in the configuration file.

5 Installing DM-MP

5.1 RedHat Linux 4

Ensure that the Device Mapper and the Multipath tools packages are installed on the system.

For RHEL4 the packages are (*check for later versions*):

```
device-mapper-1.02.25-2.e14
device-mapper-multipath-0.4.5-31.e14
```

You must configure the host to start the Device Mapper Multipath daemon, `multipathd`, at boot time. Installing the package will not make this happen.

5.1.1 Multipath Setup

Complete the following steps on your RHEL4 host to start the `multipathd` daemon at boot time:

1. Run the following command to determine whether the daemon is configured to start:

```
# chkconfig --list multipathd
```

2. If is not configured, run the following commands to start the `multipathd` daemon during boot time:

```
# chkconfig --add multipathd
# chkconfig multipathd on
```

For more information please see `CHKCONFIG(8)`.

3. After completing the above steps reboot the host. To verify that the modules required by DM-MP are loaded and the `multipathd` daemon has started at boot time, run the following commands:

```
# lsmod | grep dm_multipath
# lsmod | grep dm_mod
# ps ax | grep multipathd
```

5.2 RedHat Linux 5

For RedHat 5 the packages are (*check for later versions*):

```
device-mapper-1.01-1.6
multipath-tools-0.4.5-0.11
```

You must configure the host to start the Device Mapper Multipath daemon, `multipathd`, at boot time. Installing the package will not make this happen. Refer to instructions for RedHat 4.

5.3 SUSE Linux 9

For SUSE 9 the packages are (*check for later versions*):

```
device-mapper-1.01.01-1.6
multipath-tools-0.4.7-75.21
```

You must configure the host to start the Device Mapper Multipath daemon, `multipathd`, at boot time. Installing the package will not make this happen. Refer to instructions for RedHat 4.

5.4 SUSE Linux 10

For SUSE 10 the packages are (*check for later versions*):

```
device-mapper-1.02.13-6.9
multipath-tools-0.4.7-34.18
```

You must configure the host to start the Device Mapper Multipath daemon, `multipathd`, at boot time. Installing the package will not make this happen. Refer to instructions for RedHat 4.

6 Configuration

6.1 Configuring the DotHill storage array for Multipathing

Multipath configurations require that you map each instance of a storage resource (partition) to two different host channels on the SANnet controller. This is explained in detail in the *SANnet II 200 FC and SATA Array Best Practices Manual* (part number 83-00003263) available in the Dot Hill Web site (www.dothill.com/support).

6.2 Configuring DM-MP with the **multipath.conf** file

Initial configuration of DM-MP devices is done through entries in the configuration file `/etc/multipath.conf`. The parameters in this file provide a variety of options for setting the attributes and behavior of a multipath device including the I/O spreading policy, the path selector algorithm, the polling interval for path status verification, and the program used by multipath to obtain the WWID.

After making changes to the `multipath.conf` file you must restart the `multipathd` daemon for the changes to take effect. Table 1 lists the important attributes appropriate for the SANnet and R/Evo arrays along with their descriptions and possible values.

Table 1: Supported Multipath Attributes

Attribute	Description	Supported Values
<code>path_grouping_policy</code>	Determines how paths are assembled into path groups.	<code>failover</code>: one path per path group <code>multibus</code>: all valid paths in one path group <code>group_by_serial</code>: one path group per detected controller <code>group_by_prio</code>: one path group per path priority value
<code>path_checker</code>	Command that is used to determine the status of a path.	<code>tur</code>: SCSI TEST UNIT READ command <code>readsector0</code>: READ sector 0 of storage resource
<code>getuid_callout</code>	Command that is used to obtain the WWID for the device	<code>scsi_id -g -u -s device</code>
<code>path_selector</code>	Specifies the algorithm used for path selection within a path group.	<code>round-robin 0</code>
<code>failback</code>	Manages the path group failback	<code>manual</code> <code>immediate</code> <code>n (n>0)</code>: failback deferred for n seconds
<code>prio_callout</code>	Executable to obtain a path weight for a block device. Weights are summed for each path group to determine the next path group to use in case of path failure	<code>none</code> <code>/sbin/mpath_prio_alua</code>

no_path_retry (only for RHEL4)	<i>n</i> is the number of retries until disable queuing (queues till <i>n</i> number of polling), or <i>fail</i> means immediate failure (no queuing), <i>queue</i> means never stop queuing (queue forever or until the path comes alive)	• <i>n</i> (>0) • <i>fail</i> • <i>queue</i>
features (only for SLES)	Sets the SuSE DM-MP driver to queue I/O if no path is available. This is recommended for SuSE 9.0.	• “1 queue_if_no_path”

Each of these attributes can be set either on a per-device basis for a single storage resource, on a per-storage array basis of for all the storage resources presented by one model of storage array, or for all storage resources seen by the system. The file is divided into four sections: system defaults, blacklisted devices, per-storage array settings, and per-device settings. Settings made in the per- device section apply only to the device specified in that section; settings made in the per-storage array section apply to all devices presented by that model of storage array. Devices in the blacklisted devices section are excluded from DM-MP control.

To determine the attributes of a multipath device, the blacklisted devices section is checked first. If the device does not match an entry in the blacklist, then the per-device section is checked. If there is no match there, then the per-storage array section is checked. Devices that do not match entries in any of the previous three sections inherit the settings in the system defaults section.

In addition to a template file, `/etc/multipath.conf`, the DM-MP package installs three other versions of this configuration file:

- `/usr/share/doc/device-mapper-multipath-0.4.5/multipath.conf.annotated`
This file contains detailed descriptions of each of the parameters that can be used. (A copy of this file is included as an appendix to this document for easy reference.)
- `/usr/share/doc/device-mapper-multipath-0.4.5/multipath.conf.defaults`
This file contains the settings that are compiled into the code. Running DM-MP without a configuration file is the same as using the `multipath.conf.defaults` file.
- `/usr/share/doc/device-mapper-multipath-0.4.5/multipath.conf.synthetic`.
This file contains the same entries as the template file that is placed into the `/etc` directory during package installation

6.3 Multipath.conf (Additional Detail)

6.3.1 System defaults (`defaults`)

Devices that are not specified in one of the other three sections of the file inherit their properties from the defaults section. The following settings are compiled into the code and are also show in the `multipath.conf.defaults` file:

```

defaults {
    udev_dir                /dev
    polling_interval        10
    path_selector            "round-robin 0"
    path_grouping_policy     multibus
    getuid_callout           "/sbin/scsi_id -g -u -s /block/%n"
    prio_callout             /bin/true
    path_checker             readsector0
    rr_min_io               100
    rr_weight               priorities
    failback                immediate
    no_path_retry            fail
    user_friendly_names     yes
}

```

6.3.2 Blacklisted devices (devnode_blacklist)

The blacklisted devices section provides various options for specifying devices that are to be exempt from Device Mapper Multipath control. Devices can be specified by either their device node name (devnode) or their WWID (wwid). The following lines exclude from a SCSI disk with the specified WWID and all IDE devices. Two other commented lines are shown as examples.

```

blacklist {
    wwid SSEAGATE_ST318406LC_____3FE0QZN20000222895FJ
    devnode "^hd[a-z]"
    # devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    # devnode "^cciss!c[0-9]d[0-9]*"
}

```

Using devnode names of the form `sdX` should be avoided since the devices to which the reference can change as storage is reconfigured.

6.3.3 Per-storage-array (devices)

Settings in the per-storage-array section are used for all multipath devices presented by storage arrays whose SCSI VENDOR ID and PRODUCT ID match entries in one of the device sections. In the following example we set different attributes for devices presented by two different models of Dot Hill storage arrays:

```
devices {
    device {
        vendor            "DotHill"
        product           "SANnet II FC"
        path_grouping_policy multibus
        failback           immediate
    }
    device {
        vendor            "DotHill"
        product           "R/Evo 2730-2R"
        path_grouping_policy failover
        failback           25
    }
}
```

6.3.4 Per-device settings (multipaths)

The per-device settings specify one or more devices by WWID. These settings apply only to the devices of a specific WWID value. This gives the user an opportunity to customize the behavior of DM-MP for each storage resource. One useful feature is the ability to specify a customized alias for a device. Since the association between the WWID and the storage resource is permanent, a alias assigned by this method can never change. For example, the following multipaths section assigns names of “myspace” and “yourspace” to two of the storage resources:

```
multipaths {
    multipath {
        wwid            3600c0ff0000000000023432491536d600
        alias           myspace
        failback         manual
    }
    multipath {
        wwid            3600c0ff00000000000234320d5321e101
        alias           yourspace
        failback         45
    }
}
```

This sets the aliases of the multipath devices to my chosen names rather than the default user-friendly names of the `mpathn` format.

6.3.5 Unified Lun (ULP) settings (devices)

The per-device settings here are used when using a DotHill Storage product with Unified Luns. The important settings for ULP performance are the “path_grouping_policy” and the “prio_callout” flags. This example is taken from the DotHill R/Evolution 2330, which is an iSCSI storage array.

```
devices {
    device {
        vendor                "DotHill"
        product               "R/Evo 2330-2R"
        path_grouping_policy  group_by_prio
        getuid_callout        "/sbin/scsi_id -g -u -s /block/%n"
        path_checker          tur
        path_selector         "round-robin 0"
        hardware_handler      "0"
        prio_callout          " /sbin/mpath_prio_alua /dev/%n"
        failback              immediate
        rr_weight             uniform
        no_path_retry         12
    }
}
```

6.4 Multipath.conf (Recommended DotHill Storage Settings)

6.4.1 SANnet II

SANnet II - The following are the recommended attributes for the SANnet II arrays specified in the per-storage-array section:

```

devices {
    device {
        vendor                "DotHill"
        product                "SANnet II FC"
        path_grouping_policy   multibus
        getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
        path_checker           tur
        path_selector          "round-robin 0"
        hardware_handler       "0"
        prio_callout           none
        failback               15
        rr_weight              priorities
        no_path_retry          queue
    }
}

```

6.4.2 R/Evolution 2730 / 5730

R/Evolution 2730 and 5730 Series (Non-ULP)- The following are the recommended attributes for the R/Evolution 2730 and 5730 arrays, without ULP, specified in the per-storage-array section:

```

devices {
    device {
        vendor                "DotHill"
        product                "R/Evolution 2730-2R"
        #product               "R/Evolution 5730-2R"
        path_grouping_policy   multibus
        getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
        path_checker           tur
        path_selector          "round-robin 0"
        hardware_handler       "0"
        prio_callout           none
        failback               immediate
        rr_weight              priorities
        no_path_retry          queue
    }
}

```

6.4.3 R/Evolution 5730 (with ULP)

R/Evolution 5730 Series (ULP)- The following are the recommended attributes for the R/Evolution 5730 array, with ULP, specified in the per-storage-array section:

```

devices {
    device {
        vendor                "DotHill"
        product                "R/Evolution 5730-2R"
        path_grouping_policy   group_by_prio
        getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
        path_checker           tur
        path_selector          "round-robin 0"
        hardware_handler       "0"
        prio_callout           " /sbin/mpath_prio_alua /dev/%n"
        failback               immediate
        rr_weight              priorities
        no_path_retry          12
    }
}

```

6.4.4 R/Evolution 2330

R/Evolution 2330 (Non-ULP) - The following are the recommended attributes for the R/Evolution 2330 array, without ULP, specified in the per-storage

```

devices {
    device {
        vendor                "DotHill"
        product                "R/Evo 2330-2R"
        path_grouping_policy   multibus
        getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
        path_checker           tur
        path_selector          "round-robin 0"
        hardware_handler       "0"
        prio_callout           none
        failback               immediate
        rr_weight              uniform
        no_path_retry          12
    }
}

```

6.4.5 R/Evolution 2330 (with ULP)

R/Evolution 2330 (ULP)- The following are the recommended attributes for the R/Evolution 2330 array, with ULP, specified in the per-storage

```
devices {
    device {
        vendor                "DotHill"
        product                "R/Evo 2330-2R"
        path_grouping_policy   group_by_prio
        getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
        path_checker           tur
        path_selector          "round-robin 0"
        hardware_handler       "0"
        prio_callout           " /sbin/mpath_prio_alua /dev/%n"
        failback               immediate
        rr_weight              uniform
        no_path_retry          12
    }
}
```

6.4.6 R/Evolution 2530

R/Evolution 2530 (ULP)- The following are the recommended attributes for the R/Evolution 2530 array, with ULP, specified in the per-storage

```
devices {
    device {
        vendor                "DotHill"
        product                "R/Evo 2530-2R"
        path_grouping_policy   group_by_prio
        getuid_callout         "/sbin/scsi_id -g -u -s /block/%n"
        path_checker           tur
        path_selector          "round-robin 0"
        hardware_handler       "0"
        prio_callout           " /sbin/mpath_prio_alua /dev/%n"
        failback               immediate
        rr_weight              uniform
        no_path_retry          12
    }
}
```

7 Configuring Host Bus Adapters

Ensure that the current Fibre Channel HBA driver is installed on the system.

You must configure HBA parameters for the Device Mapper Multipath. The HBA time-out values are typically set for non-DM-MP environments, where longer time outs are necessary. This is because the only alternative is to send an error to the application. However, with DM-MP, errors like cable failures must be reported earlier so that the multipath layer can quickly take action and redirect the I/O to another path.

This section describes how to configure the host bus adapter driver:

7.1 Configuring QLogic HBA parameters

To configure QLogic HBA parameter, complete the following steps:

1. For QLogic 2xxx family of HBAs, add the following line to the `/etc/modprobe.conf` file:

```
options qla2xxx qlport_down_retry=1 ql2xretrycount=5 ql2xfailover=0
```

2. Rebuild the `initrd` file using the `mkinitrd` command. A sample command is show below. For more information see `MKINITRD(8)`.

```
# mkinitrd /boot/initrd-2.6.9-34ELsmp-dmmpparams 2.6.9-34.ELsmp
```

3. Edit your boot loader configuration file to make your new `initrd` file available and reboot the host.

7.2 Configuring QLogic/Emulex HBAs for R/Evolution storage

Please refer to the *REvo 2730 Interoperability Matrix* and *Revolution Storage System Best Practices Guide* (part number 83-00004171) for setting HBA parameters for use against the R/Evo storage. These documents are available in the Dot Hill Web site (www.dothill.com/support).

8 Managing DM-MP Devices

Initial configuration of multipath devices is done by the `multipath.conf` file. However, the command line tool provides a way to make modifications to their attributes as well as monitor them. Table 2 lists the basic options supported by the multipath CLI utility.

Table 2: Basic Command of Device Mapper Multipath CLI

Command	Description
<code>multipath</code>	Create DM-MP devices
<code>multipath -d</code>	Display potential paths but do not create any devices
<code>multipath -F</code>	Delete all DM-MP devices
<code>multipath -l [device]</code>	Display the current multipath topology from sysfs and device mapper

<code>multipath -ll [device]</code>	Display the current multipath topology from all available sources (including path checkers)
<code>multipath -v2 [device]</code>	Re-scan path information
<code>multipath -v3</code>	Display all device information including all paths, source of attributes (e.g., default)
<code>multipath -p group-policy [<device>]</code> <code>multipath [<device>]</code>	Set group policy to specified policy (e.g., multipath)

To view the current status of all multipath devices, run the multipath utility:

```
# multipath -ll
mpath1 (3600c0ff000000000234320d5321e100)
[size=25 GB][features="0"][hwhandler="0"]
\_ round-robin 0 [prio=2]
  \_ 2:0:0:0      sdb 8:16 [ready]
  \_ 3:0:0:0      sde 8:64 [ready]

mpath2 (3600c0ff000000000234320d5321e101)
[size=25 GB][features="0"][hwhandler="0"]
\_ round-robin 0 [prio=2]
  \_ 2:0:0:1      sdc 8:32 [ready]
  \_ 3:0:0:1      sdf 8:80 [ready]
```

The information in the map is presented by grouping the paths for storage resource with unique identifiers. The size, features and the corresponding hardware handlers are also shown:

alias (WWID if different from alias) [size] [features] [hardware handler]

A summary line is displayed for each path group:

`\_ scheduling_policy path_group_priority [path_group_status]`

After that follow a line for each path in the path group:

`\_host:channel:ID:LUN devnode major:minor [path_status] [dm_status]`

At various time the status of a path may not be know.

The following display shows devices with custom names (values of alias) that were set in the configuration file. The presence of a single path in each path group is a result of the attribute failover for the path_grouping_policy:

```
# multipath -ll
yourspace (3600c0ff0000000000234320d5321e101)
[size=25 GB][features="0"][hwhandler="0"]
\_ round-robin 0 [prio=1][active]
  \_ 2:0:0:1      sdc 8:32 [active][ready]
\_ round-robin 0 [prio=1][enabled]
  \_ 3:0:0:1      sdf 8:80 [active][ready]

myspace (3600c0ff000000000023432491536d600)
[size=33 GB][features="0"][hwhandler="0"]
\_ round-robin 0 [prio=1][active]
  \_ 2:0:1:0      sdd 8:48 [active][ready]
\_ round-robin 0 [prio=1][enabled]
  \_ 3:0:1:0      sdg 8:96 [active][ready]
```

The multipath devices are created in the `/dev/mapper` directory. These devices can be used like any other block devices present in the host for any block- or file-level I/O operations.

The state of the path is given as `[ready]` `[active]` if the path is up and ready for I/O. If the path is down, this state will be shown as `[faulty]` `[failed]`. The path states will be updated periodically by `multipathd` based on the polling interval specified in `/etc/multipath.conf`.

If you want to reduce the number assigned to the `/dev/mapper/mpath` link, you can edit the “`/var/lib/multipath/binding`” file. Just delete the uncommented `wwid/alias` lines in the file.

9 Known Issues

Following are the limitations and known issues of Device Mapper Multipath:

- `ioctl` support is not available on a multipath device created by DM-MP. Hence SCSI inquiry like operations are not possible with these devices
- You have to manually enter the device properties for the SANnet arrays in the `/etc/multipath.conf` file.
- Under all path failure condition, if the `queue_if_no_path` feature is enabled, the I/O requests are queued as long as system resources are available. This may result in system resource starvation.
- Wild card or regular expression support for the product and vendor string (in the `/etc/multipath.conf` file) is only supported in RHEL 4 U3
- Time based I/O request queuing (`no_path_retry` in the `/etc/multipath.conf` file) is only supported in RHEL4U3.
- Device Mapper Multipath may fail to create device nodes for partitions when the configuration has a large number of LUNs
- For a host with many LUNs, the command `multipath -ll` may take longer time to display the status of all the DM-MP devices if there is a change in the path status
- Device Mapper Multipath does not support online deletion of multipath devices. If the

storage resource is deleted from the host when it is online, the DM-MP device corresponding to that LU is displayed as `###:###:###` in the `multipath -ll` output.

- Online LUN addition is currently not supported
- The current version supports only a single path to the root device
- SLES 9.3 “*user_friendly_names*” may not create names with *mpathN*. Instead, edit the `multipath.conf` file with the `wwid` of the device.
- When iSCSI is the protocol, the “*no_path_retry*” flag should be set to *12* and not *queue*. There have been issues with Linux servers freezing when using *queue* as the value.

10 Appendix A - multipath.conf.annotated file

The annotated multipath.conf file describes all of the supported options and the defaults that are compiled into the code. This file is installed in the /usr/share/doc/device-mapper-multipath-0.4.5/ directory. It is reproduced here as a reference:

```
/usr/share/doc/device-mapper-multipath-0.4.5/multipath.conf.annotated
##
## This is a template multipath-tools configuration file
## Uncomment the lines relevant to your environment
##
#
##
## name : defaults
## desc : multipath-tools default settings
##
#defaults {
#    #
#    # name      : udev_dir
#    # desc      : directory where udev creates its device nodes
#    # default   : /udev
#    #
#    udev_dir    /dev
#
#    #
#    # name      : polling_interval
#    # scope     : multipathd
#    # desc      : interval between two path checks in seconds
#    # values    : n > 0
#    # default   : 5
#    #
#    polling_interval 10
#
#    #
#    # name      : selector
#    # scope     : multipath
#    # desc      : the default path selector algorithm to use
#    #           : these algorithms are offered by the kernel multipath target
#    # values    : "round-robin 0"
#    # default   : "round-robin 0"
#    #
#    selector    "round-robin 0"
```

```

#
#
#   # name      : path_grouping_policy
#   # scope     : multipath
#   # desc      : the default path grouping policy to apply to unspecified
#   #             multipaths
#   # values    : failover          = 1 path per priority group
#   #             multibus          = all valid paths in 1 priority group
#   #             group_by_serial   = 1 priority group per detected serial
#   #             number
#   #             group_by_prio     = 1 priority group per path priority
#   #             value
#   #             group_by_node_name = 1 priority group per target node name
#   # default    : failover
#   #
#   path_grouping_policy    multibus
#
#
#   # name      : getuid_callout
#   # scope     : multipath
#   # desc      : the default program and args to callout to obtain a unique
#   #             path identifier. Absolute path required
#   # default    : /sbin/scsi_id -g -u -s
#   #
#   getuid_callout  "/sbin/scsi_id -g -u -s /block/%n"
#
#
#   # name      : prio_callout
#   # scope     : multipath
#   # desc      : the default program and args to callout to obtain a path
#   #             priority value. The ALUA bits in SPC-3 provide an
#   #             exploitable prio value for example. "none" is a valid value
#   # default    : (null)
#   #
#   #prio_callout   "/bin/true"
#
#
#   # name      : path_checker
#   # scope     : multipath & multipathd
#   # desc      : the default method used to determine the paths' state
#   # values    : readsector0|tur|emc_clariion|hp_sw|directio
#   # default    : readsector0
#   #
#   path_checker    readsector0
#
#

```

```
#      # name      : rr_min_io
#      # scope     : multipath
#      # desc      : the number of IO to route to a path before switching
#      #            to the next in the same path group
#      # default   : 1000
#      #
#      rr_min_io    100
#
#      #
#      # name      : rr_weight
#      # scope     : multipath
#      # desc      : if set to priorities the multipath configurator will assign
#      #            path weights as "path prio * rr_min_io"
#      # values    : priorities|uniform
#      # default   : uniform
#      #
#      rr_weight    priorities
#
#      #
#      # name      : failback
#      # scope     : multipathd
#      # desc      : tell the daemon to manage path group failback, or not to.
#      #            0 means immediate failback, values >0 means deferred
#      #            failback expressed in seconds.
#      # values    : manual|immediate|n > 0
#      # default   : immediate
#      #
#      failback     manual
#
#      #
#      # name      : no_path_retry
#      # scope     : multipath & multipathd
#      # desc      : tell the number of retries until disable queueing, or
#      #            "fail" means immediate failure (no queueing),
#      #            "queue" means never stop queueing
#      # values    : queue|fail|n (>0)
#      # default   : (null)
#      #
#      no_path_retry queue
#
#      #
#      # name      : user_friendly_names
#      # scope     : multipath
#      # desc      : If set to "yes", using the bindings file
#      #            /var/lib/multipath/bindings to assign a persistent and
#      #            unique alias to the multipath, in the form of mpath<n>.
```

```

#           #           If set to "no" use the WWID as the alias. In either case
#           #           this be will be overridden by any specific aliases in this
#           #           file.
#           # values   : yes|no
#           # default  : no
#           user_friendly_names no
#
#}
#
##
## name      : blacklist
## scope     : multipath & multipathd
## desc      : list of device names to discard as not multipath candidates.
##            Devices can identified by either their device node name "devnode"
##            or their WWID "wwid".
## default   : cciss, fd, hd, md, dm, sr, scd, st, ram, raw, loop
##
#blacklist {
#   wwid 26353900f02796769
#   devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
#   devnode "^hd[a-z]"
#   devnode "^cciss!c[0-9]d[0-9]*"
#}
#
##
## name      : multipaths
## scope     : multipath & multipathd
## desc      : list of multipaths finest-grained settings
##
#multipaths {
#   #
#   # name      : multipath
#   # scope     : multipath & multipathd
#   # desc      : container for settings that apply to one specific multipath
#   #
#   multipath {
#       #
#       # name      : wwid
#       # scope     : multipath & multipathd
#       # desc      : index of the container
#       #
#       wwid          3600508b4000156d700012000000b0000
#
#       #
#       # name      : alias
#       # scope     : multipath

```

```

#           # desc      : symbolic name for the multipath
#           #
#           alias                yellow
#
#           #
#           # name       : path_grouping_policy
#           # scope      : multipath
#           # desc       : path grouping policy to apply to this multipath
#           # values     : failover          = 1 path per priority group
#           #               multibus         = all valid paths in 1 priority
#           #               group
#           #               group_by_serial  = 1 priority group per detected
#           #               serial number
#           #               group_by_prio    = 1 priority group per path
#           #               priority value
#           #               group_by_node_name = 1 priority group per target
#           #               node name
#           # default    : failover
#           #
#           path_grouping_policy    multibus
#
#           #
#           #
#           # name       : path_selector
#           # desc       : the path selector algorithm to use for this mpath
#           #               these algo are offered by the kernel mpath target
#           # values     : "round-robin 0"
#           # default    : "round-robin 0"
#           #
#           path_selector           "round-robin 0"
#
#           #
#           # name       : failback
#           # scope      : multipathd
#           # desc       : tell the daemon to manage path group failback, or
#           #               not to. 0 means immediate failback, values >0 means
#           #               deffered failback expressed in seconds.
#           # values     : manual|immediate|n > 0
#           # default    : immediate
#           #
#           failback                manual
#
#           #
#           # name       : rr_weight
#           # scope      : multipath
#           # desc       : if set to priorities the multipath configurator will

```

```

#           #           assign path weights as "path prio * rr_min_io"
#           # values   : priorities|uniform
#           # default  : uniform
#           #
#           rr_weight      priorities
#
#           #
#           # name      : no_path_retry
#           # scope     : multipath & multipathd
#           # desc      : tell the number of retries until disable queueing,
#           #           or "fail" means immediate failure (no queueing),
#           #           "queue" means never stop queueing
#           # values    : queue|fail|n (>0)
#           # default   : (null)
#           #
#           no_path_retry  queue
#       }
#   multipath {
#       wwid      1DEC_____321816758474
#       alias     red
#       rr_weight      priorities
#   }
#}
#
##
## name   : devices
## scope  : multipath & multipathd
## desc   : list of per storage controller settings
##         overrides default settings (device_maps block)
##         overridden by per multipath settings (multipaths block)
##
#devices {
#   #
#   # name   : device
#   # scope  : multipath & multipathd
#   # desc   : settings for this specific storage controller
#   #
#   device {
#       #
#       # name   : vendor, product
#       # scope  : multipath & multipathd
#       # desc   : index for the block
#       #
#       vendor      "COMPAQ   "
#       product     "HSV110 (C)COMPAQ"
#   }
#}

```

```

#
#
#       # name      : path_grouping_policy
#       # scope     : multipath
#       # desc      : path grouping policy to apply to multipath hosted
#                   by this storage controller
#       # values    : failover          = 1 path per priority group
#                   multibus          = all valid paths in 1 priority
#                   group
#                   group_by_serial    = 1 priority group per detected
#                   serial number
#                   group_by_prio      = 1 priority group per path
#                   priority value
#                   group_by_node_name = 1 priority group per target
#                   node name
#       # default   : failover
#
#       path_grouping_policy    multibus
#
#
#       # name      : getuid_callout
#       # scope     : multipath
#       # desc      : the program and args to callout to obtain a unique
#                   path identifier. Absolute path required
#       # default   : /sbin/scsi_id -g -u -s
#
#       getuid_callout          "/sbin/scsi_id -g -u -s /block/%n"
#
#
#       # name      : prio_callout
#       # scope     : multipath
#       # desc      : the program and args to callout to obtain a path
#                   weight. Weights are summed for each path group to
#                   determine the next PG to use case of failure.
#                   "none" is a valid value.
#       # default   : no callout, all paths equals
#
#       prio_callout            "/sbin/mpath_prio_balance_units %d"
#
#
#       # name      : path_checker
#       # scope     : multipathd
#       # desc      : path checking algorithm to use to check path state
#       # values    : readsector0, tur
#       # default   : readsector0
#
#       path_checker            readsector0

```

```
#
#
#       #
#       # name      : path_selector
#       # desc      : the path selector algorithm to use for this mpath
#       #             these algo are offered by the kernel mpath target
#       # values    : "round-robin 0"
#       # default   : "round-robin 0"
#       #
#       path_selector          "round-robin 0"
#
#       #
#       # name      : failback
#       # scope     : multipathd
#       # desc      : tell the daemon to manage path group failback, or
#       #             not to. 0 means immediate failback, values >0 means
#       #             deffered failback expressed in seconds.
#       # values    : manual|immediate|n > 0
#       # default   : immediate
#       #
#       failback              30
#
#       #
#       # name      : hardware_handler
#       # scope     : multipath
#       # desc      : If set, it specifies a module that will be used to
#       #             perform hardware specific actions when switching
#       #             path groups or handling IO errors
#       # values    : "0"|"1 emc"
#       # default   : "0"
#       #
#       hardware_handler "1 emc"
#
#       #
#       # name      : rr_weight
#       # scope     : multipath
#       # desc      : if set to priorities the multipath configurator will
#       #             assign path weights as "path prio * rr_min_io"
#       # values    : priorities|uniform
#       # default   : uniform
#       #
#       rr_weight            priorities
#
#       #
#       # name      : no_path_retry
#       # scope     : multipath & multipathd
#       # desc      : tell the number of retries until disable queueing,
```

```
#           #           or "fail" means immediate failure (no queueing),
#           #           "queue" means never stop queueing
#           # values   : queue|fail|n (>0)
#           # default  : (null)
#           #
#           no_path_retry queue
#       }
#   device {
#       vendor           "COMPAQ   "
#       product          "MSA1000   "
#       path_grouping_policy multibus
#       path_checker      tur
#       rr_weight          priorities
#   }
#}
```